Clustering Complex Zeros of Triangular System of Polynomials

Rémi Imbach, Marc Pouget, Chee Yap

Clustering Complex Zeros of Triangular System of Polynomials*

Rémi Imbach[1], Marc Pouget[2], and Chee Yap[1]

[1] Courant Institute of Mathematical Sciences, New York University, USA
remi.imbach@nyu.edu, yap@cs.nyu.edu
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
marc.pouget@inria.fr

Abstract. This paper gives the first algorithm for finding a set of natural ϵ-clusters of complex zeros of a triangular system of polynomials within a given polybox in $\mathbb{C}^n$, for any given $\epsilon > 0$. Our algorithm is based on a recent near-optimal algorithm of Becker et al (2016) for clustering the complex roots of a univariate polynomial where the coefficients are represented by number oracles.
Our algorithm is numeric, produces guaranteed results and is based on subdivision. We implemented it and compared it with state of the art solvers on various triangular systems, including systems with clusters of solutions or multiple solutions.

Keywords: complex root finding · triangular polynomial system · near-optimal root isolation · certified algorithm · complex root isolation · oracle multivariable polynomial · subdivision algorithm · Pellet's theorem.

1 Introduction

This report considers the fundamental problem of finding the complex solutions of a system $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ of n polynomial equations in n complex variables $\mathbf{z} = (z_1, \ldots, z_n)$. We will call *triangular* a system $\mathbf{f} = (f_1, \ldots, f_n) : \mathbb{C}^n \to \mathbb{C}^n$ where $f_i \in \mathbb{C}[z_1, \ldots, z_i]$ for $1 \le i \le n$. Throughout this paper, we use boldface symbols to denote vectors and tuples; for instance $\mathbf{0}$ stands for $(0, \ldots, 0)$.

We are interested in finding clusters of solutions of triangular systems and in counting the total multiplicity of solutions in clusters. Solving triangular systems is a fundamental task in polynomial equations solving, since many algebraic approaches (Gröbner basis, CAD, resultants,...) generally reduce the original system to triangular systems.

* Rémi's work is supported by the European Union's Horizon 2020 research and innovation programme No. 676541, NSF Grants # CCF-1563942, # CCF-1564132 and # CCF-1708884. Chee's work is supported by NSF Grants # CCF-1423228 and # CCF-1564132.

The problem of isolating the complex solutions of a polynomial system in an initial region-of-interest (ROI) is defined as follows: let $\mathtt{Zero}(\mathbf{B}, \mathbf{f})$ denote the set of solutions of $\mathbf{f}$ in $\mathbf{B}$, regarded[3] as a multiset.

Local Isolation Problem (LIP):
Given: a polynomial map $\mathbf{f} : \mathbb{C}^n \to \mathbb{C}^n$, a polybox $\mathbf{B} \subset \mathbb{C}^n$, $\epsilon > 0$
Output: a set $\{\boldsymbol{\Delta}^1, \ldots, \boldsymbol{\Delta}^l\}$ of pairwise disjoint polydiscs of radius $\leq \epsilon$ where
- $\mathtt{Zero}(\mathbf{B}, \mathbf{f}) = \bigcup_{j=1}^{l} \mathtt{Zero}(\boldsymbol{\Delta}^j, \mathbf{f})$.
- each $\mathtt{Zero}(\boldsymbol{\Delta}^j, \mathbf{f})$ is a singleton.

This is "local" because we restrict attention to roots in a ROI $\mathbf{B}$. There are two issues with (LIP) as formulated above: deciding if $\mathtt{Zero}(\boldsymbol{\Delta}^j, \mathbf{f})$ is a singleton, and deciding if such a singleton lies in $\mathbf{B}$, are two "zero problems" that require exact computation. Generally, this can only be decided if $\mathbf{f}$ is algebraic. Even in the algebraic case, this may be very expensive. In [5,1] these two issues are side-stepped for the univariate case by defining the local clustering problem which is described next in a multivariate setting.

Before proceeding, we fix some notations. A *polydisc* $\boldsymbol{\Delta}$ is a vector of complex discs. The *center* (resp. *radius* $r(\boldsymbol{\Delta})$) of $\boldsymbol{\Delta}$ is the vector of the centers (resp. radii) of its components. If δ is any positive real number, we denote by $\delta\boldsymbol{\Delta}$ the polydisc that has the same center than $\boldsymbol{\Delta}$ and radius $\delta r(\boldsymbol{\Delta})$. We also say $r(\boldsymbol{\Delta}) \leq \delta$ if each component of $r(\boldsymbol{\Delta})$ is $\leq \delta$. A *(square complex) box* B is a complex interval $[\ell_1, u_1] + \boldsymbol{i}([\ell_2, u_2])$ where $u_2 - \ell_2 = u_1 - \ell_1$ and $\boldsymbol{i} := \sqrt{-1}$; the *width* $w(B)$ of B is $u_1 - \ell_1$ and the *center* of B is $u_1 + \frac{w(B)}{2} + \boldsymbol{i}(u_2 + \frac{w(B)}{2})$. A *polybox* $\mathbf{B} \in \mathbb{C}^n$ is a vector boxes. The *center* (resp. *width* $w(\mathbf{B})$) of $\mathbf{B}$ is the vector of the centers (resp. widths) of its components. If δ is any positive real number, we denote by $\delta\mathbf{B}$ the polybox that has the same center than $\mathbf{B}$ and width $\delta w(\mathbf{B})$.

We introduce three notions to define the local solution clustering problem. Let $\mathbf{a} \in \mathbb{C}^n$ be a solution of $\mathbf{f}(\mathbf{z}) = \mathbf{0}$. The *multiplicity* of $\mathbf{a}$ in $\mathbf{f}$, also called the *intersection multiplicity* of $\mathbf{a}$ in $\mathbf{f}$ is classically defined by localization of rings as in [6, Def. 1, p. 61], we denote it by $\#(\mathbf{a}, \mathbf{f})$. An equivalent definition uses dual spaces, see [3, Def. 1, p. 117]. For any set $S \subseteq \mathbb{C}^n$, we denote by $\mathtt{Zero}(S, \mathbf{f})$ the multiset of zeros of $\mathbf{f}$ in S, and $\#(S, \mathbf{f})$ the total multiplicity of $\mathtt{Zero}(S, \mathbf{f})$. If S is a polydisc so that $\mathtt{Zero}(S, \mathbf{f})$is non-empty, we call $\mathtt{Zero}(S, \mathbf{f})$ a *cluster* and S an *isolator* of the cluster. If in addition, we have that $\mathtt{Zero}(S, \mathbf{f}) = \mathtt{Zero}(3 \cdot S, \mathbf{f})$, we call $\mathtt{Zero}(S, \mathbf{f})$ a *natural cluster* and call S a *natural isolator*. In the context of numerical algorithm, the notion of cluster of solutions is more meaningful than that of solution with multiplicity since the perturbation of a multiple solution

[3] A *multiset* S is a pair $(\underline{S}, \mu)$ where $\underline{S}$ is an ordinary set called the *underlying set* and $\mu : \underline{S} \to \mathbb{N}$ assigns a positive integer $\mu(x)$ to each $x \in \underline{S}$. Call $\mu(x)$ the *multiplicity* of x in S, and $\mu(S) := \sum_{x \in \underline{S}} \mu(x)$ the *total multiplicity* of S. Also, let $|S|$ denote the cardinality of $\underline{S}$. If $|S| = 1$, then S is called a *singleton*. We can form the union $S \cup S'$ of two multisets with underlying set $\underline{S} \cup \underline{S'}$, and the multiplicities add up as expected.

generates a cluster. We thus "soften" the problem of isolating the solutions of a triangular system of polynomial equations while counting their multiplicities by translating it into the local solution clustering problem defined as follows:

Local Clustering Problem (LCP):
Given: a polynomial map $\mathbf{f} : \mathbb{C}^n \to \mathbb{C}^n$, a polybox $\mathbf{B} \subset \mathbb{C}^n$, $\epsilon > 0$
Output: a set of pairs $\{(\boldsymbol{\Delta}^1, m^1), \ldots, (\boldsymbol{\Delta}^l, m^l)\}$ where:
- the $\boldsymbol{\Delta}^j$s are pairwise disjoint polydiscs of radius $\leq \epsilon$,
- each $m^j = \#(\boldsymbol{\Delta}^j, \mathbf{f}) = \#(3\boldsymbol{\Delta}^j, \mathbf{f})$
- $\mathtt{Zero}(\mathbf{B}, \mathbf{f}) \subseteq \bigcup_{j=1}^{l} \mathtt{Zero}(\boldsymbol{\Delta}^j, \mathbf{f}) \subseteq \mathtt{Zero}(2\mathbf{B}, \mathbf{f})$.

In this (LCP) reformulation of (LIP), we have removed the two "zero problems" noted above: we output clusters to avoid the first one, and we allow the output to contain zeroes outside the ROI $\mathbf{B}$ to avoid the second one. We choose $2\mathbf{B}$ for simplicity; it is easy to replace the factor of 2 by $1 + \delta$ for any desired $\delta > 0$.

2 Our contributions

We propose an algorithm for solving the (LCP), *i.e.* computing natural clusters, for a triangular system $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ with a zero-dimensional solution set. To this end, we propose a formula to count the sum of multiplicities of solutions in a cluster. Our formula is derived from a result of [6] that links the intersection multiplicity of a solution of a triangular system to multiplicities in fibers. We define *towers of clusters* to encode clusters of solutions of a triangular system in stacks (or towers) of clusters of roots of univariate polynomials.

Our algorithm leverages from the triangular form of $\mathbf{f}$: it computes first clusters of solutions of $f_1 = \cdots = f_{n-1} = 0$, then clusters of roots of f_n on fibers over clusters previously found. The components of those fibers are clusters of roots of univariate polynomials that are advantageously represented by *oracles*; oracle means here a procedure providing approximations at any precision.

We propose a bound on the loss of accuracy induced by the specialization of f_n on a fiber approximated at a given precision; the coefficients of f_n specialized in oracle fibers are thus also known by oracles.

To compute clusters of roots of a univariate polynomial given as an oracle, we rely on the recent algorithm described in [1], based on a predicate introduced in [2] that combines Pellet's theorem and Graeffe iterations to determine the number of roots counted with multiplicities in a complex disc; this predicate is called *soft* because it only requires the polynomial to be known as approximations. It is used in a subdivision framework combined with Newton iterations to achieve a near optimal complexity. An implementation of [1] is described in [4].

We also implemented our algorithm and made this implementation available to the community[4] as a package for `Julia`[5]. We compare our algorithm with state of the art solvers for solving triangular algebraic systems. We also consider

[4] https://github.com/rimbach/Ccluster.jl
[5] https://julialang.org/

the case of triangular systems obtained by triangularization with regular chains algorithms. Our implementation appears to be particularly efficient for systems with a small number of variables but with high degrees. Another interesting feature is the ability to solve locally, that is over a small polybox.

References

1. Becker, R., Sagraloff, M., Sharma, V., Xu, J., Yap, C.: Complexity analysis of root clustering for a complex polynomial. In: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation. pp. 71–78. ISSAC '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2930889.2930939
2. Becker, R., Sagraloff, M., Sharma, V., Yap, C.: A near-optimal subdivision algorithm for complex root isolation based on Pellet test and Newton iteration. J. Symbolic Computation **86**, 51–96 (May-June 2018)
3. Dayton, B.H., Zeng, Z.: Computing the multiplicity structure in solving polynomial systems. In: Proceedings of the 2005 international symposium on Symbolic and algebraic computation. pp. 116–123. ACM (2005)
4. Imbach, R., Pan, V.Y., Yap, C.: Implementation of a near-optimal complex root clustering algorithm. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) Mathematical Software – ICMS 2018. pp. 235–244. Springer International Publishing, Cham (2018)
5. Yap, C., Sagraloff, M., Sharma, V.: Analytic root clustering: A complete algorithm using soft zero tests. In: The Nature of Computation. Logic, Algorithms, Applications. LNCS, vol. 7921, pp. 434–444. Springer (2013)
6. Zhang, Z., Fang, T., Xia, B.: Real solution isolation with multiplicity of zero-dimensional triangular systems. Science China Information Sciences **54**(1), 60–69 (2011)